

The connectivity of graphs of graphs with self-loops and a given degree sequence

JOEL NISHIMURA

School of Mathematical and Natural Sciences
Arizona State University, Glendale, AZ 85306-4908, USA

Abstract

‘Double edge swaps’ transform one graph into another while preserving the graph’s degree sequence, and have thus been used in a number of popular Markov chain Monte Carlo (MCMC) sampling techniques. However, while double edge-swaps can transform, for any fixed degree sequence, any two graphs inside the classes of simple graphs, multigraphs, and pseudographs, this is not true for graphs which allow self-loops but not multiedges (loopy graphs). Indeed, we exactly characterize the degree sequences where double edge swaps cannot reach every valid loopy graph and develop an efficient algorithm to determine such degree sequences. The same classification scheme to characterize degree sequences can be used to prove that, for all degree sequences, loopy graphs are connected by a combination of double and triple edge swaps. Thus, we contribute the first MCMC sampler that uniformly samples loopy graphs with any given sequence.

1 Introduction

Understanding what properties of an empirical graph are noteworthy, as opposed to those which are merely the consequence of the degree sequence, is often addressed by comparing the empirical graph with an ensemble of sampled graphs with the same degree sequence [14, 15]. While uniformly sampling graphs with a fixed degree sequence seems straightforward, it can be surprisingly complex. How one samples and the resulting graph statistics are dependent on the space of graphs considered: e.g. whether self-loops and/or multiedges are considered, and whether graphs with distinct ‘stub-labelings’ are considered unique [10].

Graphs which allow self-loops can arise in many disparate applications. For example, self-loops may represent: an author citing themselves; a protein capable of interacting with itself [12, 19]; gene operon self-regulation [17]; cannibalism in a food web [20]; users on photo sharing site Flickr linking to themselves; a loop road or cul-de-sac in a road network [7]; a repeated word in a word adjacency network; traffic flow inside an autonomous system on the Internet [13], along with many other possible interpretations. Considering networks which may include self-loops can be important both because self-loops are often of interest themselves, and because the inclusion of self-loops effectively reduces the number of edges that aren’t self-loops, potentially affecting many different network statistics, especially in small networks. Moreover, while it is commonly thought that self-loops are asymptotically rare [3], they only are for particular assumptions on degree sequences, and are more rare in ‘stub labeled’ spaces as thoroughly detailed in [10, 5]. In contrast, a so called ‘vertex-labeled’ graph is much more likely to have self-loops, yet techniques for sampling from this

space are largely undeveloped. This paper discusses loopy graphs, graphs where each vertex can have at most a single self-loop and edges are either present or absent (i.e. no multiedges).

For many different types of graphs, one of the most popular sampling techniques is Markov chain Monte Carlo sampling via ‘double edge swaps’. Guarantees about the uniformity of double edge swap MCMC sampling (along with a related family MCMC techniques [4, 6]) are founded on several properties, the most difficult of which is whether an MCMC sampler can sample every possible graph, or equivalently, whether the associated Markov chain is irreducible (equivalently, the associated graph of the Markov chain is strongly connected). For any degree sequence, the following spaces are connected and thus can be sampled using MCMC techniques: simple graphs [21, 2, 1, 9, 18], simple connected graphs [18, 2], multigraphs [11] and multigraphs with self-loops [8]. Absent from this list is the space of loopy graphs. Indeed, for some degree sequences, the standard MCMC approach applied to the space of loopy graphs cannot sample all possible such graphs. In this paper, we investigate which degree sequences have disconnected Markov chains, developing an algorithm that can detect this disconnectivity and prove that augmenting the standard MCMC ‘double edge swap’ with ‘triple edge swaps’ guarantees the chain is connected for all degree sequences. These techniques allow the space of loopy graphs to be used in the study of empirical networks.

2 The graph of loopy-graphs

Consider a graph with self-loops $G = (V, E)$, with n vertices in vertex set V and edge set E , which may or may not include self-loops: edges of the form (u, u) . Notice that loopy-graphs include simple graphs as a special case. As opposed to multigraphs, edges can appear at most once in E . For a vertex u , we denote the set of adjacent vertices, or ‘neighbors’ of u , as $N(u)$, and we refer to $|N(u)|$ as the degree of vertex of u . We adopt the convention that each self-loops contributes two to a node’s degree.

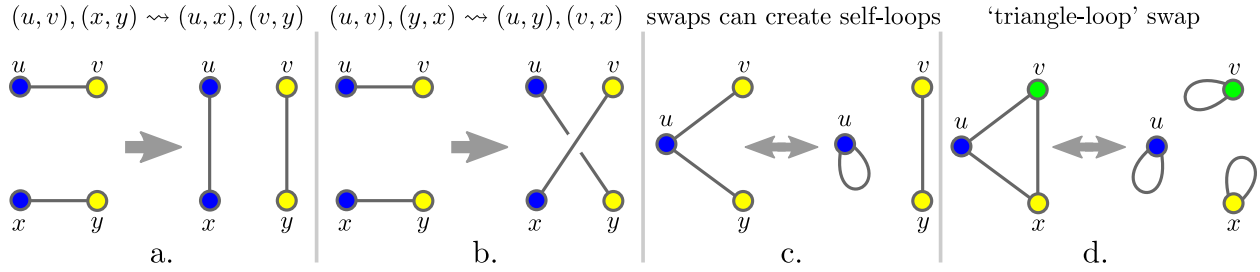


Figure 1: (a, b) For any pair of edges there are two possible double edge swaps. (c) Swapping adjacent edges creates self-loops, and swaps which involve a single self-loop can remove it. (d) In Theorem 6.1 we show that including a triple edge swap in addition to double edge swaps leads to a connected graph of loopy graphs \mathcal{G}_Δ .

Transforming one graph into another with the same degree sequence is possible through a double edge swap [16], where, as in figure 2, swapping edges (u, v) and (x, y) replaces those edges with (u, x) and (v, y) , a process we denote as $(u, v), (x, y) \rightsquigarrow (u, x), (v, y)$. Repeatedly performing double edge swaps (resampling the current graph whenever a proposed swap would create a multiedge) is the basis of MCMC samplers of loopy graphs. Conceptually, repeatedly performing double edge swaps is a random walk on a graph whose vertices are loopy graphs, with the same prescribed degree sequence. Specifically, for a given degree sequence $\{k_i\}$ let $\mathcal{G}(\{k_i\})$ be the *graph of loopy*

graphs under double edge-swaps where each vertex is a loopy graph and edge (G_i, G_j) exists if and only if there is a single double edge swap that takes G_i to G_j . Showing that a MCMC sampler can sample from graphs with degree sequence $\{k_i\}$ requires showing that $\mathcal{G}(\{k_i\})$ is connected, otherwise random walks will not be able to reach all possible loopy-graphs.

In fact though, the space of loopy-graphs is not connected under double edge-swaps for every possible degree sequence. In section 3, we introduce two classes of graphs, Q_1 and Q_2 such that if \mathcal{G} contains a loopy-graph $G \in Q_1 \cup Q_2$, then \mathcal{G} is disconnected. Both classes Q_1 and Q_2 require high degree nodes, and in section 5 we discuss tests to determine whether a given degree sequence can create a graph in Q_1 or Q_2 .

Moreover, Q_1 and Q_2 exactly characterize the graphs which cause \mathcal{G} to be disconnected, as shown in the first main theorem, Theorem 4.20 in section 4. In contrast, any two graphs not in Q_1 or Q_2 are connected with each other, and this will be established by studying special maximal elements of \mathcal{G} . The general outline is as follows: from any graph G_i , let \hat{G}_i be the graph connected to G_i in \mathcal{G} with the maximum number of self-loops (along with a few other technical considerations), as in Figure 2. Next, inside a graph \hat{G}_i , let V_i^0 be the subset of vertices that do not have self-loops, let V_i^k be the vertices that have a shortest path of length k to a node in V_i^0 , and let V^∞ be the nodes disconnected to V^0 . Based on the largest clique $K^0 \subseteq V^0$, we classify the structure of \hat{G}_i as one of five different types, (four types are displayed in Figure 4), two of which (\hat{G}^3 and \hat{G}^d , $d > 3$) belong to Q_1 and Q_2 .

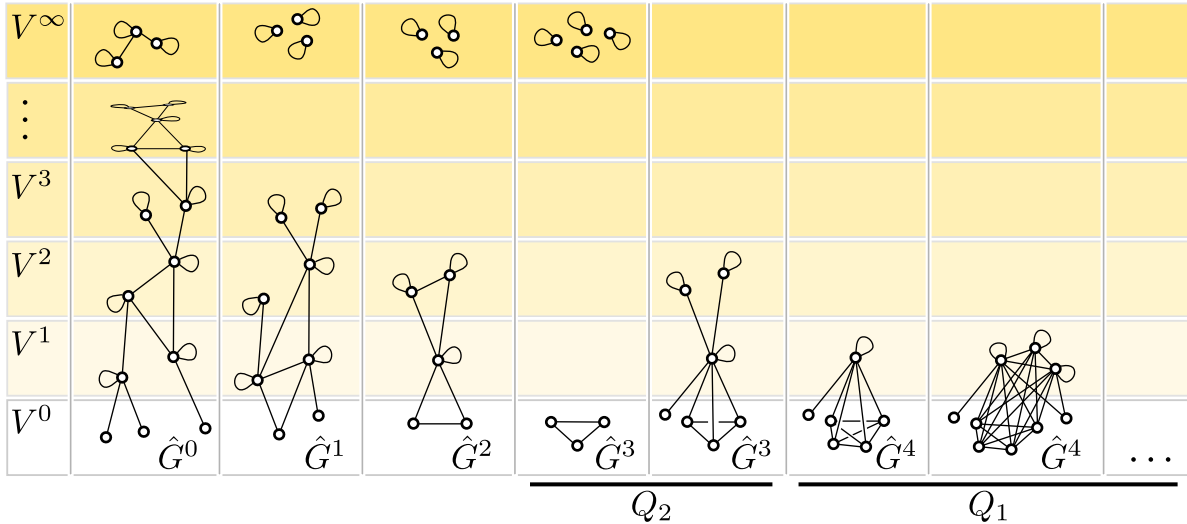


Figure 2: Each of the above graphs has the maximal number of self-loops accessible through double edge swaps. Vertices are categorized by their distance to a vertex without a self-loop and graphs are labeled by the size of the largest clique in V^0 . Graphs with cliques of size 3 or greater in V^0 are of class Q_1 or Q_2 , as shown by theorem 4.19.

The categorization of possible structures of \hat{G} suggests Algorithm 1 which determines whether a degree sequence has a connected or disconnected \mathcal{G} . Another consequence of Theorem 4.20 is that any degree sequence $\{k_i\}$ that is disconnected, is disconnected because there are graphs with triangles which cannot be changed into self-loops, which naturally suggests Theorem 6.1, which states that the space of graphs with self-loops is connected under the combination of double and triple edge swaps. Based on this theorem we suggest a MCMC approach that uniformly samples graphs with self-loops and a fixed degree sequence.

3 Degree sequences with disconnected \mathcal{G}

First we consider a simple disconnected case, which establishes that for some degree sequences \mathcal{G} is not connected.

3.1 Cycles and cliques

The simplest example of a degree sequence that is not connected is $\{2, 2, 2\}$, which can be wired either as a triangle, or as 3 self-loops. Since there are no valid double edge swaps of either the triangle graph or 3 self-loops (all swaps would create multiedges) the space is disconnected. The disconnectivity of $\{2, 2, 2\}$ can be extended in two ways, to larger cycles with and to larger cliques. The degree sequence of a cycle, $\{2, 2, \dots, 2\}$ clearly has a disconnected space, since a graph composed only of nodes with self-loops has no valid double edge swaps. Similarly a clique with additional self-loops at up $n - 3$ vertices has alternate configurations, but lacks any valid double edge swaps, implying that the degree sequence: $\{n + 1, \dots, n + 1, n - 1, \dots, n - 1, n - 1\}$ is also disconnected.

As a useful exercise, we consider the structure of $\mathcal{G}(\{2, 2, \dots, 2\})$ in more detail. Any graph with degree sequence $\{2, 2, \dots, 2\}$ is composed of isolated self-loops and cycles of length at least 3. Further, any valid double edge swap either:

1. creates a self-loop and reduces a k cycle, $k \geq 4$, to a $k - 1$ cycle (swapping adjacent edges);
2. combines a self-loop with a k cycle to create a $k + 1$ cycle (swapping a self-loop and an edge in a cycle);
3. merges two cycles into a larger cycle (swapping edges in separate cycles);
4. cuts a cycle into two smaller cycles, each with length at least 3 (non-adjacent edges in the same cycle);
5. swaps two edges in the same cycle without changing its length (non-adjacent edges in the same cycle).

If double edge swaps are augmented with a triple edge swap that takes a triangle to three self-loops (and another triple edge swap that does the reverse), then it is clear that every graph in the space can be taken to the graph made entirely of self-loops (and thus \mathcal{G} is connected) via the following procedure:

1. by swapping edges in different cycles, combine all cycles into a single long cycle;
2. from the graph's one cycle, swap adjacent edges to create self-loops until the single cycle has length 3;
3. use a triple edge swap to replace the only length 3 cycle with 3 self-loops.

3.2 Other disconnected graphs

These disconnected examples will be generalized into two classes of graphs Q_1 and Q_2 , displayed in Figure 3, which generalize the problems with the clique and the cycle respectively. In section 4 we show that Q_1 and Q_2 describe all disconnected graphs.

Definition 3.1 (Q_1). *A graph G is of class Q_1 when the following conditions are true of G :*

1. *There exists a clique K^0 in V^0 with $|K^0| \geq 4$ (recall: V^0 is the set of nodes without self-loops)*

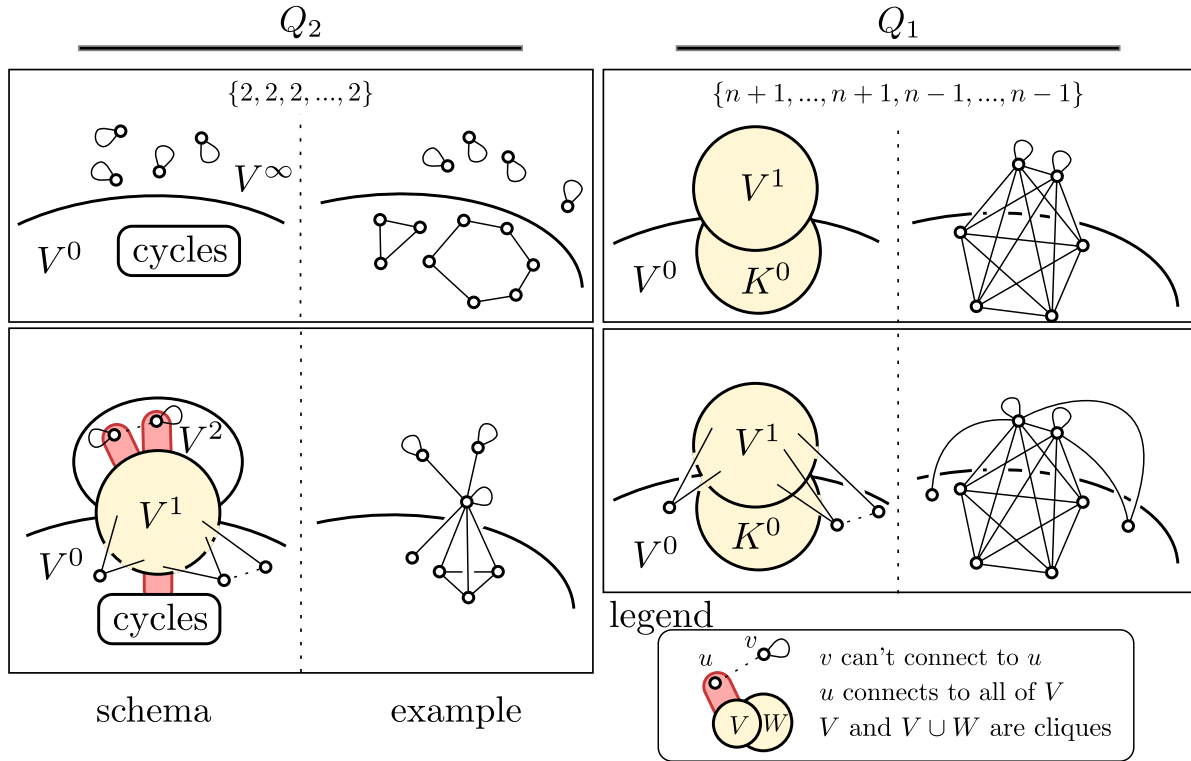


Figure 3: Both the degree sequence $\{n+1, \dots, n+1, n-1, \dots, n-1\}$ and $\{2, 2, \dots, 2\}$ have a disconnected \mathcal{G} whose disconnectivity can be generalized to classes Q_1 and Q_2 . The schematic for Q_2 includes $\{2, 2, \dots, 2\}$ as a special case if when V^1 is empty, V^2 is relabeled as V^∞ .

2. For any $u \in V^0$, either u has no neighbors in V^0 or u is in the clique K^0 ,
3. $V^1 \cup K^0$ is a clique,
4. $V^2 = V^\infty = \emptyset$.

We will later show that all \hat{G}^d , $d > 3$ are of class Q_1 . The important feature of Q_1 is that it is closed under any double edge swap.

Lemma 3.2. *For any two graphs G_1 and G_2 connected via a double edge swap, if $G_1 \in Q_1$ then $G_2 \in Q_1$.*

Proof. The structure of Q_1 implies that all edges have at least one endpoint in $V^1 \cup K^0$. Since $V^1 \cup K^0$ is a clique, there are thus no valid swaps involving any edge in $V^1 \cup K^0$ as any such swap would create a multiedge. Similarly, a swap between a self-loop in V^1 and an edge from V^0 to V^1 would also create a multiedge. The only possible swaps are between two edges (u, v) and (x, y) where $u, x \in V^0 \setminus K^0$ and $v, y \in V^1$. Notice that swap $(u, v)(y, x) \rightsquigarrow (u, x), (y, v)$ is precluded by the presence of edge $(v, y) \in V^1$, while swap $(u, v)(x, y) \rightsquigarrow (u, y), (x, v)$ does not create a new edge in V^0 , alter the fact that $V^1 \cup K^0$ is a clique or create a vertex in V^2 or V^∞ . Thus Q_1 is closed under edge swaps. \square

While Q_1 includes cliques as a special case, a similar structure, Q_2 generalizes the problems associated with cycles and degree sequences $\{2, 2, 2, \dots, 2\}$. We will later show that all \hat{G}^3 are of class Q_1 .

Definition 3.3 (Q_2). *A graph G is of class Q_2 when the following conditions are true of G :*

1. There are at least three nodes in V^0 , each of node in V^0 has exactly two neighbors in V^0 (i.e. $|N(u) \cap V^0| = 2$ for $u \in V^0$).
2. For any $u \in V^0$, either, u has no neighbors in V^0 or u has exactly two neighbors in V^0 and is adjacent to all of V^1 .
3. V^1 is a clique
4. For any $u \in V^2$, $N(u) = V^1$,
5. $V^3 = \emptyset$,
6. Either V^∞ is empty or both V^1 is empty and $k_u = 2$ for $u \in V^\infty$.

Implicit in the definition of Q_2 is that there is a cycle in V^0 of length at least 3. Similarly to Q_1 , Q_2 is also closed under double edge swaps.

Lemma 3.4. *For any two graphs G_1 and G_2 connected via a double edge swap, if $G_1 \in Q_2$ then $G_2 \in Q_2$.*

Proof. If V^∞ is non-empty then properties 1 and 6 of Q_2 immediately imply that the degree sequence of non-isolated nodes is $\{2, 2, \dots, 2\}$, and this scenario was fully described earlier.

If $V^1 \neq \emptyset$, a quick check reveals that the only edge swaps that are possible (all others would require multiedges) involve swaps between two edges in V^0 , swaps between an edge in V^0 and a self-loop in V^2 , and swaps between two edges joining V^0 to V^1 . However, each of these three swaps preserves the properties of Q_2 : swaps between the two edges in V^0 rearrange the cycle structure of V^0 and potentially move a node from V^0 to V^2 , but this preserves the properties of

Q_2 ; swaps between a self-loop in V^2 and an edge in V^0 move a node from V^2 to V^0 , reversing the previous swap; For edges (u, v) and (x, y) , $u, x \in V^0$ and $v, y \in V^1$ swap $(u, v)(y, x) \rightsquigarrow (u, x), (y, v)$ is precluded by the presence of edge $(v, y) \in V^1$, while swap $(u, v)(x, y) \rightsquigarrow (u, y), (x, v)$ is only possible if both $|N(u) \cap V^0| = |N(x) \cap V^0| = 0$ and such a swap does not affect the properties of Q_2 . \square

This implies the first half of Theorem 4.20:

Corollary 3.5. *Any \mathcal{G} which contains a graph in Q_1 or Q_2 is disconnected.*

Proof. All graphs in Q_1 and Q_2 contain a closed cycle of length at least 3 in V^0 . For a graph $G \in \mathcal{G}$ and $G \in Q_1 \cup Q_2$ let C be all the cycles in V^0 . Deleting each edge in C and placing a self-loop at each node in C preserves the degree sequence and thus creates a graph $H \in \mathcal{G}$, but H does not satisfy the first criterion of either Q_1 or Q_2 . By lemmas 3.2 and 3.4, Q_1 and Q_2 are closed under double edge swaps and thus G is not connected to H . \square

Thus, we have generalized the graph of graph disconnectivity associated with cliques, as in Q_1 and cycles, as in Q_2 . As shown in the next section, Q_1 and Q_2 exactly characterize all disconnected \mathcal{G} .

4 Categorizing the components of \mathcal{G}

For any graph $G_i \in \mathcal{G}$, let $\mathcal{V}(G_i)$ be the graphs connected to G_i with the maximum number of self-loops. Of the graphs in $\mathcal{V}(G_i)$, let \hat{G}_i be a graph with the maximum number of edges contained in V^0 . While \hat{G}_i has at least as many self-loops as any other graph connected to G_i , if \mathcal{G} is not connected, \hat{G}_i may not have the maximum number of self-loops possible.

In order to formalize the meaning of a graph with the maximum number of self-loops, let $\{\bar{k}_i\}$ denote the ‘simplified degree sequence’, the degree sequence of a graph if all self-loops were deleted, (i.e. $\bar{k}_i = k_i$ for all $i \in V^0$ and $\bar{k}_i = k_i - 2$ for all $i \notin V^0$). Assuming a degree sequence $\{k_i\}$ is in an unique decreasing order let a degree sequence be m -simple-graphical if there are self-loops on the m largest degree vertices. A degree sequence being m -simple-graphical is equivalent to the condition that the following degree sequence is simple-graphical: $k_i^* = k_i - 2$ for $i \leq m$ and $k_i^* = k_i$ for $i > m$. For a degree sequence, let m^* be the maximum value of m for which the degree sequence is m -simple-graphical. We call graphs with m^* self-loops on the m^* highest degree vertices:

Definition 4.1 (m^* -loopy graphs). *A graph G is m^* -loopy if G has m^* self-loops on vertices $i \leq m^*$.*

Determining the cases where \hat{G} is m^* -loopy will be critical in the categorization of different possible \hat{G} for the following reason.

Lemma 4.2. *For $G_1, G_2 \in \mathcal{V}$, if both G_1 and G_2 are m^* -loopy then G_1 is connected to G_2 .*

Proof. Since both G_1 and G_2 are m^* -loopy they have self-loops at the same vertices and the same simplified degree sequences and thus, by the connectivity of simple graphs [18], G_1 and G_2 are connected. \square

In some degree sequences, a graph is obviously m^* -loopy because all nodes with degree at least two have self-loops. It is not always as straightforward though. For example, the degree sequences $\{4, 4, 2\}$ and $\{6, 6, 5, 3, 3, 3, 2\}$ have no configurations where all vertices have self-loops, as $\{2, 2, 0\}$ and $\{4, 4, 3, 1, 1, 1, 0\}$ are not simple-graphical degree sequences (i.e. there is not a simple graph

with those degree sequences). Instead, these graphs have valid configurations where all but the vertex with degree 2 has self-loops.

For any degree sequence there are thus two possibilities, either all graphs in \mathcal{G} are connected to m^* -loopy graphs and \mathcal{G} is connected, or there exists some graph not connected to any m^* -loopy graph and \mathcal{G} is not connected.

Understanding the possible forms of m^* -loopy graphs will comprise the majority of the remaining effort, but the simplest case may also be the most common case. For any \hat{G} where V^0 contains only vertices of degree 0 and 1, \hat{G} is clearly m^* -loopy. We now turn our attention to the much more complicated scenarios where there exists some $u \in V^0$ with $k_u \geq 2$.

The classification of the possible \hat{G} is broken up according to the size of the largest clique, $K^0 \subseteq V^0$, where \hat{G}^d has $|K^0| = d$. The critical lemmas to prove will be Lemmas 4.9, and 4.14. Lemma 4.9 states that there exists a sequence of double edge swaps which can exchange any vertex in V_0 with any other vertex of equal or lower degree. Thus any \hat{G}_i is connected to a similar graph \hat{G}_j where V_j^0 contains only the smallest degrees. Building on this, Lemma 4.14 states that a graph \hat{G}^d , $d \leq 2$, is m^* loopy. Thus, by lemma 4.2 only degree sequences that can wire a \hat{G}^d , $d \geq 3$ can be disconnected and, as will be shown in theorem 4.20, a graph $\hat{G}^3 \in Q_2$ and $\hat{G}^d \in Q_1$ for $d > 3$.

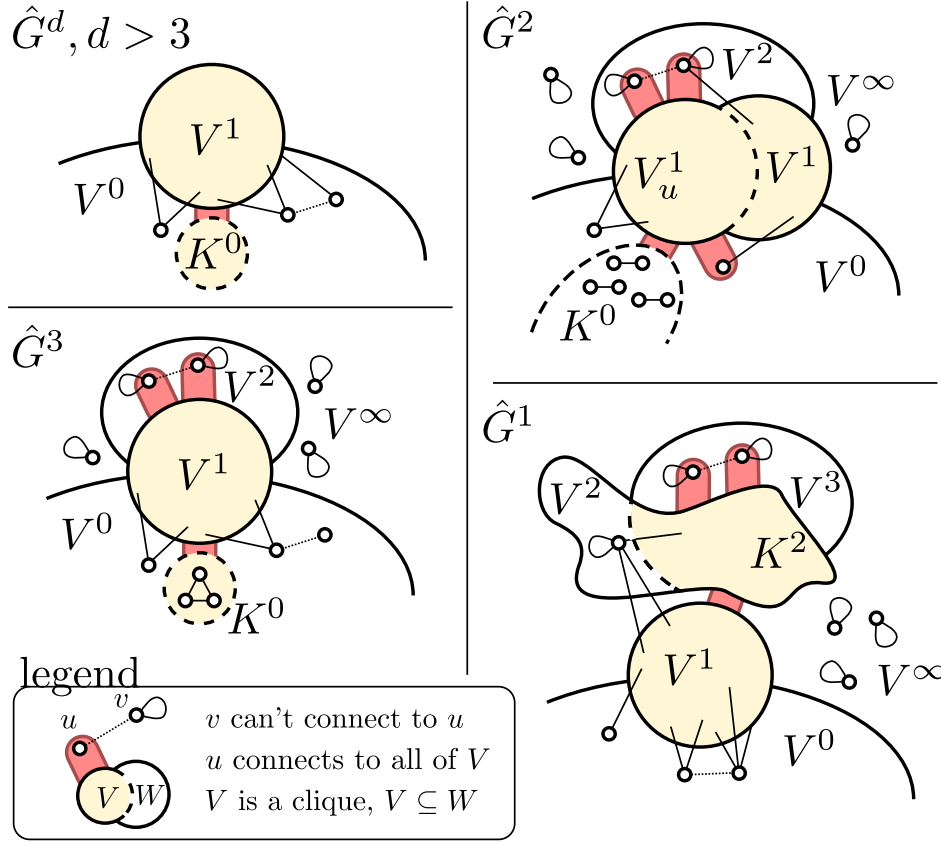


Figure 4: The size of K^0 imposes strict requirements on the possible structure of \hat{G}^d . When K^0 is a clique larger than 3 vertices ($d > 3$), \hat{G}^d is simply a clique of vertices some with self-loops, some without, and a number of vertices with connections only into members of the clique which have self-loops. Note: in \hat{G}^3 either V^∞ or V^1 must be empty.

Before proving lemmas 4.9 or 4.14 we first construct some general purpose lemmas. We begin with some investigations into restrictions on the sets V^k for all \hat{G}^d . Consider the following definition:

Definition 4.3 (Open Wedge xuv). *There exists open wedge xuv at u if there exists edges (u, v) and (u, x) and $x \notin N(v)$.*

If there exists open wedge xuv for $u \in V^0$ then the swap $(u, x), (u, v) \rightsquigarrow (u, u), (x, v)$ is possible and creates a self-loop. Since \hat{G} has the maximum number of self-loops, it must be free of: any open wedges centered at vertices in V^0 , any rewires that can create such open wedges, or any other sequence of rewires that net create self-loops.

Lemma 4.4. *For any $u \in V^0$ and $v \in N(u)$, if u connects to a vertex x , so does v .*

Proof. If not, then there exists open wedge xuv . □

Lemma 4.4 implies that any subgraph of V^0 is a clique plus isolated nodes, as is the subgraph on vertices $u \cup N(u)$ for $u \in V^0$.

Lemma 4.5. *If there exists disjoint (u, v) and (x, y) both in V^0 , $k_u \geq 2$ then $N(u) \cap V^1 = N(x) \cap V^1$*

Proof. Suppose first that $N(u) \cap V^1 \not\subseteq N(x) \cap V^1$, then there exists $w \in N(u) \cap V^1$ such that $w \notin N(x)$. If (u, x) exists, then by lemma 4.4 x must be connected to w , a contradiction. Thus (u, x) isn't present, and similarly, lemma 4.4 also implies that (u, y) , (v, x) and (v, y) aren't present. Swap $(u, v), (x, y) \rightsquigarrow (u, x), (v, y)$ is thus possible but creates open wedge wux contradicting that \hat{G} has the maximal number of self-loops.

If instead $N(x) \cap V^1 \not\subseteq N(u) \cap V^1$ then $k_x \geq 2$ and the above argument holds. □

Lemma 4.6. *If there exists $u \in V^0$, $k_u \geq 2$ then all non self-loops contain a vertex in V^0 , V^1 or V^2 .*

Proof. Suppose to the contrary that there exists (x, y) with neither x nor y in V^0 , V^1 or V^2 . For $v, w \in N(u)$, notice (v, w) must exist, otherwise there exists an open wedge vuw , swap $(v, w), (x, y) \rightsquigarrow (x, v), (y, w)$ is thus valid, but creates an open wedge at u . □

This implies that V^d is empty for all finite $d > 3$, and V^3 contains no edges asside from self-loops. This also implies that the set of vertices disconnected from V^0 can only contain isolated self-loops.

Lemma 4.7. *For \hat{G}^d with $d \geq 3$, $V^1 \subset N(u)$ for all $u \in K^0$.*

Proof. Suppose not, that there exists $x \in V^1$ independent of K^0 . Since $x \in V^1$ there exists $y \in V^0 \setminus K^0$ along with edge (x, y) . Let $u, v, w \in K^0$. Lemma 4.4 implies that since $y \notin K^0$ then y is independent of K^0 . Swapping $(u, v), (x, y) \rightsquigarrow (u, x), (v, y)$ creates an open wedge yvw . □

Lemma 4.8. *If there exists $u \in V^0$, $k_u \geq 2$ then V^1 is a clique.*

Proof. If $|V^1| = 1$, then V^1 is trivially a clique. If $|V^1| \geq 2$, then suppose to the contrary that there exists $x, y \in V^1$ such that $(x, y) \notin E$. Consider the two possible cases:

1. There exists some $u \in V^0$ such that $x, y \in N(u)$. In this case, if $(x, y) \notin E$ then there is an open wedge xuy . Thus $(x, y) \in E$.
2. There exists $u, v \in V^0$, such that (u, x) and (v, y) are in E but (u, y) and (v, x) are not. If $(u, v) \in E$ then there exists an open wedge xuv . Thus both (x, y) and (u, v) are not in E and the swap $(u, x), (v, y) \rightsquigarrow (x, y), (u, v)$ is valid, but produces a graph with one additional edge in V^0 , contradicting that \hat{G} has the maximum number of edges in V^0 .

□

Lemma 4.9. *For any vertex $x \notin V^0$, and any vertex $u \in V^0$, if $k_x \geq k_u$ then there exists a sequence of swaps that exchanges x for u in V^0 .*

Proof. First we consider the case where $x \in V^1$. For $d \in \{1, 2\}$, since V^1 is a clique, each $x \in V^1$ contains a self-loop and any $u \in V^0$ contains at most a single neighbor not in V^1 then $k_x \geq k_u$.

For $d \geq 3$, lemmas 4.7 and 4.8 imply that $K^0 \cup V^1$ is a clique, and lemma 4.4 implies that all subgraphs of V^0 are cliques, then $k_x \geq k_u + 2$ for any $x \in V^1$ and $u \in V^0$.

For $x \in V^m$, $m \geq 2$ suppose to the contrary that there exists x with degree less than u . Consider two cases, first that $N(u) \subseteq V^1$ and second that there exists edge $(u, z) \in V^0$.

1. $N(u) \subseteq V^1$: Since (x, x) contributes 2 to x 's degree, $N(u) \subseteq V^1$ and $k_x \leq k_u$ then there exists $v, w \in N(u)$ and $v, w \notin N(x)$. In such a case, notice that swap $(x, x), (v, w) \rightsquigarrow (x, v), (x, w)$ and subsequent swap $(u, v), (u, w) \rightsquigarrow (u, u), (v, w)$ exchanges x for u in V^0 .
2. There exists edge $(u, z) \in V^0$: First, swap $(x, x), (u, z) \rightsquigarrow (x, u), (x, z)$. Since $k_x \leq k_u$ and x is connected to z while $z \notin N(u)$ then there must be some $y \in N(u)$ but $y \notin N(x)$. Thus there exists open wedge xuy and swap $(x, u), (u, y) \rightsquigarrow (u, u), (x, y)$ exchanges x for u in V^0 without increasing the number of edges inside V^0 .

□

Based on lemma 4.9 we will assume WLOG that \hat{G}^d has $k_u \leq k_x$ for all $u \in V^0$ and $x \notin V^0$. It thus remains to show that lemma 4.14 is true for $d \in \{1, 2\}$ and to further restrict the possible structures when $d \geq 3$.

4.1 The structure of \hat{G}^1

Lemma 4.10. *Every \hat{G}^1 , is m^* -loopy.*

Proof. Suppose not, that there exists \hat{G}^1 which is not m^* -loopy. Since \hat{G}^1 is not m^* -loopy then there exists some simple graph $G^* = \{V^*, E^*\}$ with degree sequence equal to the simplified degree sequence of \hat{G}^1 , except at vertices in some nonempty set $S \subseteq V^0$, where $k_u^* = k_u - 2$ for $u \in S$, as in Figure 5.

For each $u \in S$ there must be at least two vertices $l_u, r_u \in N(u)$ where $l_u, r_u \notin N^*(u)$. Let $B = \bigcup_{u \in S} \{l_u \cup r_u\}$ and let $G' = \hat{G}^1$ except without self-loops and edges (u, l_u) and (u, r_u) for each $u \in S$. Notice that G' and G^* have the same degree sequence, except at vertices B , where those in G^* have a greater degree.

Let $\Omega' = E' \setminus E^*$ be the edges in G' not in G^* and let $\Omega^* = E^* \setminus E'$ be the edges in G^* not in G' . Now consider the edge disjoint cycles and paths which alternate between edges in Ω' and Ω^* . Since the degrees of all vertices in $V \setminus B$ is the same in G' and G^* , there exists a decomposition that consists entirely of alternating cycles and alternating paths beginning and ending with edges in Ω^* at vertices in B . We now consider three cases:

1. There exists an alternating cycle C , containing some edge of the form $\{(l_u, r_u)\}$: Let $C' = C \cap E'$ and $C^* = C \cap E^*$. Since the cycle is alternating, removing edges C' from \hat{G}^1 and adding edges in C^* to create a new graph G is possible and preserves the degree sequence. Further, since the graph of simple graphs is connected, there exists a sequence of double edge swaps to create G from \hat{G}^1 . However, G still contains edges (u, l_u) and (u, r_u) as these edges were precluded from set Ω' , but since (l_u, r_u) was in C' it is not in G and thus (u, l_u) and (u, r_u) form an open wedge $l_u u r_u$ contradicting the maximality of \hat{G}^1 .

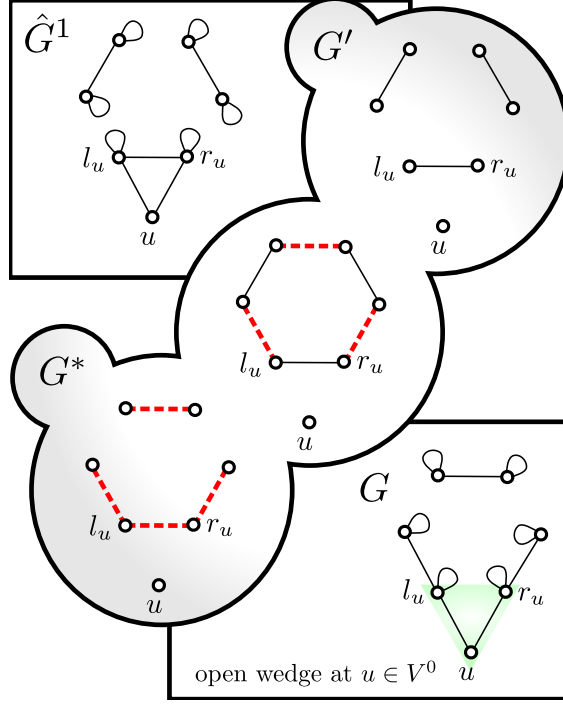


Figure 5: If there exists a graph with more self-loops than a graph \hat{G}^1 then an alternating cycle argument can show that there exists a graph G , with the same simplified degree sequence as \hat{G} but with an open wedge at $u \in V^0$.

2. There is an alternating path L beginning and ending with edges in Ω^* at nodes $u, v \in B$ where $u \neq v$: Since $B \subseteq V^1$, lemma 4.8 grants that $(u, v) \in E'$ and thus not also in Ω^* . The union $(u, v) \cup L$ produces a cycle with edges alternatingly in E' and not in E' and, as in the first case, augmenting \hat{G}^1 with this cycle produces a graph without a edge (u, v) , in violation of lemma 4.8 (Note, by classification G^1 cannot contain any edges in V^0).
3. There is an alternating path L_l beginning and ending at the same vertex $l_u \in B$ and with edges in Ω^* : Since r_u has a lower degree in G^* than in G' , there must be some alternating path L_r beginning at r_u . Further, if the second case doesn't hold, then neither L_l nor L_r can visit any other vertex in B other than l_u and r_u respectively. Let r_1 be the first vertex in path L_r . Since $(r_1, r_u) \in \Omega^*$, then by lemma 4.8 $r_1 \in V^2$. Next, if $(l_u, r_1) \in E'$, then notice that the union $(l_u, r_1) \cup L_l \cup (l_u, r_u) \cup (r_u, r_1)$ creates an alternating cycle that includes (l_u, r_u) , as in the first case. If $(l_u, r_1) \notin E'$ then the union of $(r_u, u) \cup (u, r_1) \cup L_r \setminus (r_u, r_1)$ creates an alternating cycle, and augmenting \hat{G}^1 with this cycle would create open wedge $l_u u r_1$

□

4.2 The structure of \hat{G}^2

A similar alternating path argument can be applied to \hat{G}^2 , but in some ways it's easier to investigate \hat{G}^2 directly.

First, notice that V^1 is nonempty, since a vertex $u \in V^0$ with $k_u \geq 2$ must have two neighbors but since V^0 does not contain a triangle only one of u 's neighbors can be in V^0 . For $u \in V^0$, let $V_u^1 = N(u) \cap V^1$, $V_K^1 = N(K^0) \cap V^1$ and let $n_k = |V_K^1|$.

Lemma 4.11. For \hat{G}^d , $d \geq 2$, $u \in K^0$ and any x then either $V_u^1 \subseteq N(x)$ or $N(x) \subseteq V_u^1$.

Proof. Suppose to the contrary that there exists $y \in N(x)$ and $w \in V_u^1$ but $y \notin V_u^1$ and $w \notin N(x)$. x is not connected to u as otherwise there exists open wedge xuw . If $y \in N(u)$, then it must be that $y \in V^0$ (otherwise $y \in V_u^1$) and thus there exists open wedge xyu . Thus $y \notin N(u)$.

As $d = 2$, there exists $v \in N(u) \cap K^0$. Since $y, x \notin N(u)$ then by lemma 4.4 $y, x \notin N(v)$. Now notice that swap $(x, y), (u, v) \rightsquigarrow (u, x), (v, y)$ and creates open wedge xuw . \square

Notice that this also gives that $V^3 = \emptyset$, that $k_y \geq k_x$ for any $y \in V_K^1$ and any x and, in conjunction with lemma 4.4, that $V_u^1 = V_K^1$ for any $u \in K^0$.

Lemma 4.12. For \hat{G}^d , $d \geq 2$, for $u \in K^0$ and any $x \in V^0$ then $k_u \geq k_x$.

Proof. Suppose to the contrary that there exists $x \in V^0$ with $k_x > k_u$. Since $u \in K^0$ there exists $v \in N(u) \cap V^0$. Lemma 4.4 implies that $x \notin N(u) \cup N(v)$ (otherwise $k_x = k_u$) and that there exists $y, z \in N(x)$ with $y, z \notin N(u)$. Notice that swap $(x, y), (u, v) \rightsquigarrow (x, u), (v, y)$ creates open wedge zxu , a contradiction. \square

Together with lemma 4.11, lemma 4.12 gives that all vertices $u \in V^0$ have at most one neighbor outside of V_K^1 , which is the key to the following lemma.

Lemma 4.13. Every \hat{G}^2 , is m^* -loopy.

Proof. To show this we will count the total degree inside V_K^1 , revealing it is tight with the Erdős-Gallai theorem. Consider the edges internal to V_K^1 , the edges from V^0 and the edges from all remaining vertices separately. Since $V_K^1 \subseteq V^1$, it is a clique and thus contributes $n_k(n_k - 1)$ to the degrees inside V_K^1 . Lemma 4.4 implies that for all $u \in K^0$, u connects to all of $N(K^0)$. Further, lemma 4.11 gives that for any remaining vertex x , either $N(x) \subseteq V_u^1$, in which case all k_x edges from x connect to V_K^1 , or $V_K^1 \subseteq N(x)$ in which case x connects to all of V_K^1 . Aggregating these leads to the statement

$$\sum_{u \in V_K^1} k_u = n_k(n_k - 1) + \sum_{u \in K^0} (k_u - 1) + \sum_{u \in V \setminus (K^0 \cup V_K^1)} \min(n_k, k_u) \quad (1)$$

Notice if any subset of vertices $S \in V^0$ have their degree reduced by 2 then each vertex $u \in S$ connects to at least one less vertex in V_K^1 . Thus, reducing the degree of any vertex in V^0 by 2 reduces the right side of equation 1, but not the left side, and thus by the Erdos Gallai theorem, the new degree sequence would not be simple-graphical. \square

Thus we have shown:

Lemma 4.14. A graph \hat{G}^d , $d \leq 2$, is m^* loopy.

However, as seen in Figure 2 there exists \hat{G}^d , $d \geq 3$ which are not m^* -loopy.

4.3 Structure of \hat{G}^d , $d \geq 3$

Finally we investigate the possible structures of \hat{G}^d for $d \geq 3$, showing that any \hat{G}^3 is in the classes Q_2 , any \hat{G}^d for $d > 3$ is in the class Q_1 and thus \hat{G}^d , $d \geq 3$ indicates a disconnected \mathcal{G} . First, we show the following:

Lemma 4.15. For \hat{G}^d , when $d \geq 3$ all edges in V^0 are contained in K^0 .

Proof. Suppose not, that there exists $(x, y) \in V^0$. Since $(x, y) \notin K^0$ then there is some $u \in K^0$ with $u \notin N(x)$, and since V^0 is wedge free it must be that x, y are disjoint from K^0 . Let $u, v, w \in K^0$, then swap $(x, y), (u, v) \rightsquigarrow (x, u), (y, v)$ creates wedge xuw . \square

Lemma 4.16. *For \hat{G}^d with $d > 3$ ($d = 3$), all edges (non self-loop edges) have at least one vertex in either V^0 or V^1 .*

Proof. Suppose not, that there exists (x, y) with both $x, y \notin V^0 \cup V^1$. For $u, v, w \in K^0$ notice that swapping $(u, v), (x, y) \rightsquigarrow (u, x), (v, y)$ creates open wedges xuw and yvw . If $x = y$ and there are $u, v, w, z \in K^0$ then closing wedge xuw opens a second wedge at xwz . \square

Lemma 4.17. *For \hat{G}^3 , either V^∞ or V^1 is empty.*

Proof. Suppose not, that there exists $x \in V^\infty$ and $y \in V^1$. Let $u, v, w \in K^0$. Lemma 4.7 implies that $u, v, w \in N(y)$. Consider the following swaps: $(x, x)(u, w) \rightsquigarrow (u, x)(w, x)$, then $(y, u)(u, x) \rightsquigarrow (u, u)(y, x)$ and $(v, w)(w, x) \rightsquigarrow (w, w)(v, x)$ which net creates a self-loop. \square

Lemma 4.18. *For \hat{G}^d , when $d = 3$, all vertices in V^2 have degree $|V^1| + 2$, while when $d > 3$, $V^2 = \emptyset$.*

Proof. Since $V^3 = \emptyset$ and V^2 has no internal edges by lemma 4.16, the maximum degree in V^2 is $|V^1| + 2$, while the degree of vertices in K^0 is $|V^1| + |K^0| - 1 = |V^1| + d - 1$. Since lemma 4.9 gives that $|V^1| + 2 \geq |V^1| + d - 1$ then when $d = 3$, vertices in V^2 have degree $|V^1| + 2$, and when $d > 3$, V^2 must be empty. \square

Taken together, lemmas 4.18 and 4.15 imply that \hat{G}^d for $d \geq 3$ is composed of a single large clique on $K^0 \cup V^1$ along with vertices who solely connect into that clique. Further, all vertices in V^0 have less than or equal degree than all other vertices. Meanwhile, the degrees of vertices in V^2 , and K^0 are $|V^1| + 2$ and $|V^1| + d - 2$ respectively while those in V^1 and $V^0 \setminus K^0$ have lower and upper bounds $|V^1| + |V^2| + d + 1$ and $|V^1|$ respectively. Taken together, these constraints on the form of \hat{G}^d , $d \geq 3$ (as summarized in Figure 4), can be used to detect degree sequences for which \mathcal{G} is disconnected.

Theorem 4.19. *A graph \hat{G}^3 is in the classes Q_2 while a graph \hat{G}^d for $d > 3$ is in the class Q_1 .*

Proof. In the definition of the classes Q_1 and Q_2 , the first criterion, the existence of a clique inside V^0 or of nodes with degree 2 inside V^0 are satisfied by the definitions of \hat{G}^d for $d > 3$ and \hat{G}^3 respectively. Lemma 4.15 establishes that there aren't edges in V^0 outside K^0 , and lemma 4.7 gives that any node in K^0 connects to all of V^1 ; together these satisfy the second criterion. Lemma 4.8 shows that V^1 is a clique, the third criterion of Q_2 for \hat{G}^3 , while lemmas 4.8 and 4.7 give that $V^1 \cup K^0$ is a clique for \hat{G}^d for $d > 3$, the third criterion for Q_1 . Finally, lemmas 4.18 and 4.16 imply that in \hat{G}^d for $d > 3$, $V^2 = V^\infty = \emptyset$, the fourth criterion of Q_1 . Meanwhile, in \hat{G}^3 lemma 4.16 and 4.18 imply that each $u \in V^2$ has $N(u) = V^1$, the fourth criterion of Q_2 . The last two criteria of Q_2 are satisfied by lemmas 4.17 and 4.16. Thus, $\hat{G}^3 \in Q_2$ while $\hat{G}^d \in Q_1$ for $d > 3$. \square

An immediate consequence of this along, with Lemma 4.14 and Corollary 3.5 is the following.

Theorem 4.20. *A degree sequence has a disconnected \mathcal{G} if and only if there is some graph in Q_1 or Q_2 in \mathcal{G} .*

Corollary 4.21. *Aside from the degree sequences associated with the cycle and the clique all simple graphical degree sequences have a connected \mathcal{G} .*

Proof. Applying the Erdős-Gallai theorem to the set V^1 in a graph in Q_1 or Q_2 reveals that such a graph's degree sequence is not simple-graphical, unless $|V^1| = 0$, in which case the graph is either a clique, or has degree sequence $\{2, 2, 2, \dots, 2\}$. \square

Thus, many of the most commonly examined degree sequences have a connected \mathcal{G} . However, in the space of loopy-graphs, there are many possible degree sequences which are loopy-graphical, but not simple-graphical (for example, those in Figure 2). In this next section, we discuss several ways to detect if a loopy-graphical degree sequence has a connected or disconnected space.

5 Detecting connectivity in \mathcal{G}

For many applications, detecting if a degree sequence is not at risk of being disconnected can be achieved simply by examining the maximum degree. Let n^* be the number of nodes with nonzero degree in a degree sequence $\{k_i\}$.

Theorem 5.1. *For degree sequence $\{k_i\} \neq \{2, 2, 2, \dots, 2\}$, if $\max_i k_i < 2\sqrt{n^* - 3} + 1$ then $\mathcal{G}(\{k_i\})$ is connected.*

Proof. Since only degree sequences that can wire graphs in Q_1 and Q_2 have a disconnected graph of graph, we need only show that the maximum degree of graphs in Q_1 and Q_2 is never less than $\sqrt{n^* - 3} + 2$. For a graph $G \in Q_1 \cup Q_2$, let $\alpha = |V^1|$. Notice that the highest degree node in G must be in V^1 . Counting the edges into V^1 : at least three nodes in K^0 connect to all nodes in V^1 and the remaining $n^* - 3 - \alpha$ nodes have at least one edge into V^1 . Since V^1 is a clique, there are at least $\alpha(\alpha - 1) + 3\alpha + (n^* - 3 - \alpha)$ edge endpoints into V^1 , and thus the maximum degree of a node in V^1 must be at least $\alpha + 1 + \frac{n^* - 3}{\alpha}$. Minimizing this over α yields the bound $2\sqrt{n^* - 3} + 1$ \square

When the maximum degree is larger than the bound in theorem 5.1, the following procedure can exactly identify all degree sequences aside from $\{k_i\} = \{2, 2, 2, \dots, 2\}$ and $\{k_i\} = \{n - 1, n - 1, \dots, n - 1\}$ which can wire a Q_1 or Q_2 are detected by the following procedure:

1. delete the vertex with minimum degree k_i
2. reduce the largest k_i degrees by 1
3. if all remaining degrees have two different values $\{a, b\} \geq 3$, $n_a \geq 3$, $b - 2 = n_a + n_b - 1$ and $a - 2 = n_b$ then it is possible to place nodes with degree b in V^1 , three vertices with degree a into K^0 and the remaining vertices with degree a as vertices in V^2 , creating a graph \hat{G}^3 .
4. if all remaining degrees have two different values $\{a, b\} \geq 3$, $n_a \geq 3$, $a = b - 2$ and $a = n_a + n_b - 1$ then it is possible to wire this into a clique with self-loops at each of b , creating a graph \hat{G}^d , $d \geq 4$.

Stated more formally, this procedure leads to Algorithm 1.

Theorem 5.2. *For any degree sequence, Algorithm 1 correctly identifies whether \mathcal{G} is connected or disconnected.*

Proof. To see the correctness of Algorithm 1, consider it applied to a graph with the structures \hat{G}^d for $d = 3$ and $d > 3$ in Figure 4, where the structure in Figure 4 is established in the preceding lemmas. First, notice that the algorithm directly tests for $\{2, 2, \dots, 2\}$ and $\{n - 1, n - 1, \dots, n - 1\}$.

If a degree sequence can construct a \hat{G}^d for $d \geq 3$ then $V^0 \setminus K^0$ contains only the smallest degrees and these all connect to vertices in V^1 which have the largest degrees. Further, notice that

Algorithm 1 Attempt non m^* -loopy wiring

Require: degree sequence $\{k_i\}$ **Ensure:** a non m^* -loopy graph \hat{G} , otherwise *False*

```
 $n = |\{k_i | k_i > 0\}|$ 
 $G \leftarrow$  graph initialized with vertices from  $\{k_i\}$ 
sort  $\{k_i\}$  in decreasing order
if  $n \leq 2$  then
    return False
end if
if  $\min_i k_i = \max_i k_i = 2$  then
    return a cycle graph on  $n$  vertices
end if
if  $\min_i k_i = \max_i k_i = n - 1$  then
    return a clique on  $n$  vertices
end if
for  $j \in 0 : n$  do
    if  $\{k_i\}$  has exactly two unique values then
         $a \leftarrow \min_i k_i$ 
         $b \leftarrow \max_i k_i$ 
         $n_a \leftarrow$  number of occurrences of  $a$  in  $\{k_i\}$ 
         $n_b \leftarrow$  number of occurrences of  $b$  in  $\{k_i\}$ 
         $n_t \leftarrow n_a + n_b$ 
        if  $a \geq 3$  and  $n_a \geq 3$  then
            if  $a = b - 2$  and  $a = n_t - 1$  then
                for  $u \in 0 : n_b$  do
                    add edge  $(u, u)$  to  $G$ 
                end for
                add clique on vertices  $0 : n_t$  to  $G$ 
                return  $G$ 
            end if
            if  $b - 2 = n_t - 1$  and  $a - 2 = n_b$  then
                for  $u \in 0 : n_b$  do
                    for  $v \in 0 : n_t$  do
                        add edge  $(u, v)$  to  $G$ 
                    end for
                end for
                for  $u \in 0 : (n_t - 3)$  do
                    add edge  $(u, u)$  to  $G$ 
                end for
                add clique on vertices  $(n_t - 3) : n_t$  to  $G$ 
                return  $G$ 
            end if
        end if
         $MinInd \leftarrow n - j - 1$ 
         $MinDeg \leftarrow k_{MinInd}$ 
        delete( $k_{MinInd}$ )
        if  $MinDeg > |\{k_i\}|$  then
            return False
        end if
        for  $y \in 0 : MinDeg$  do
             $k_y \leftarrow k_y - 1$ 
            add edge  $(MinInd, y)$  to  $G$ 
```

for two edges (u, v) and (x, y) with $u, x \in V^0 \setminus K^0$ and $v, y \in V^1$, swap $(u, v), (x, y) \rightsquigarrow (u, y), (x, v)$ exchanges u and x 's neighbors in V^1 . Thus, there is some \hat{G}^d for which deleting the smallest degree vertices and subtracting one from the largest remaining degrees, as in Algorithm 1, is precisely sequentially deleting actual vertices and edges from \hat{G}^d .

Once all vertices in $V^0 \setminus K^0$ have been deleted, the remaining vertices are either in K^0 , in V^1 and if $d = 3$ possibly also in V^2 but with the same degree as vertices in K^0 . Thus, the entire remaining degree sequence is composed of just two values, a and b with $a < b$. Let n_a denote the number of occurrences of degree a and n_b the number of occurrences of b . If $d = 3$ then vertices in K^0 and V^2 connect to themselves or two other vertices and all of V^1 , giving that $a = n_b + 2$; while vertices in V^1 connect to all remaining vertices, implying $b = n_a + n_b - 1$. If $d > 3$ then V^0 and V^1 form a clique and so $b = a + 2$ and $a = n_a + n_b - 1$. Thus, any degree sequences that produce an m^* loopy graph are correctly identified and conversely.

Conversely, when the algorithm returns a graph, analyzing the steps of the algorithm in reverse reveals that it has indeed constructed a valid \hat{G}^d graph with the specified degree sequence. Thus, Algorithm 1 only returns valid non m^* -loopy graphs. \square

6 Sampling loopy-graphs

A small change to \mathcal{G} can connect the space. For distinct u, v, w consider the following triple edge swap, the ‘triangle-loop’ swap, $(u, v), (v, w), (w, u) \rightsquigarrow (u, u), (v, v), (w, w)$ along with its reverse $(u, u), (v, v), (w, w) \rightsquigarrow (u, v), (v, w), (w, u)$. Let \mathcal{G}_Δ be the graph \mathcal{G} but with additional edges connecting graphs which are separated by a single triangle-loop swap.

Theorem 6.1. \mathcal{G}_Δ is connected.

Proof. Since every \hat{G}^d , $d \geq 3$ contains a triangle in V^0 , no such graph has the maximal number of self-loops. Thus for every degree sequence, any graph is connected to a \hat{G}^d for $d \leq 2$, which are m^* -loopy and since all m^* -loopy graphs are connected, the space is thus connected. \square

This allows for an MCMC sampler of the uniform distribution of graphs in \mathcal{G}_Δ . For a given degree sequence, if Algorithm 1 indicates that \mathcal{G} is connected then the standard double edge swap MCMC in [10] suffices. On the other hand, if Algorithm 1 returns a valid \mathcal{G}^d for $d \geq 3$ then triangle-loop swaps are required to connect the space, as in the procedure in Algorithm 2. Stated succinctly the stub-labeled version does the following:

From any graph G : with probability $\epsilon > 0$ pick three edges from G , if possible perform a triangle-loop swap, otherwise resample G ; with probability $1 - \epsilon$ pick 2 edges at random, if possible perform a double edge swap, otherwise resample G .

For any $\epsilon > 0$, theorem 6.1 gives that this procedure will be able to reach all graphs in \mathcal{V} , however, since the majority of proposed triple swaps will not result in a new graph, the value of ϵ that produces the optimal mixing time is likely small. In order to see that triangle-loop swaps preserve the regularity of the \mathcal{G}_Δ , notice that since each triangle-loop swap is reversible, that at any graph G each of the exactly $\binom{m}{3}$ sets of three distinct edges corresponds to an incoming edge, either from a self-loop, or a valid triangle-loop swap.

To see that \mathcal{G}_Δ is aperiodic consider several cases. Notice that for any degree sequence, if $|\mathcal{V}| \geq 2$, \mathcal{G}_Δ must contain a graph with at least one of the following: a triangle, an open wedge, two self-loops or two independent edges. Attempting to rewire two sides of a triangle or two self-loops would create a self-loop, and this attempted swap corresponds to a self-loop in \mathcal{G}_Δ , which implies that \mathcal{G}_Δ is aperiodic. Any graph with an open wedge or two independent edges has a sequence of

Algorithm 2 MCMC step (labeled stubs)

Require: loopy-graph G , stubs_labels $\in [True, False]$

Ensure: a loopy-graph adjacent to G in \mathcal{G}_Δ

```
if  $Unif(0, 1) < \epsilon$  then
    choose three edges at random
    if edges create a triangle or are self-loops & triangle-loop wouldn't create multiedges then
        perform triangle-loop swap
    end if
else
    choose two edges  $e_1$  and  $e_2$  at random
    if double edge swap wouldn't create multiedges then
        if neither  $e_1$  or  $e_2$  is a self-loop OR not stubs_labels then
            perform double edge swap
        else
            if  $Unif(0, 1) < \frac{1}{2}$  then
                perform double edge swap
            end if
        end if
    end if
end if
return  $G$ 
```

three double-edge swaps which return to the same graph, this combined with the reversible nature of double-edge swaps implies that \mathcal{G}_Δ is aperiodic.

This leads to the following theorem, which lets us conclude that Algorithm 2 forms the basis for a MCMC sampler of loopy-graphs.

Theorem 6.2. *A random walk on \mathcal{G}_Δ has a uniform stationary distribution.*

Proof. As an aperiodic, regular, connected graph \mathcal{G}_Δ has a unique uniform stationary distribution. \square

7 Conclusion

By examining the possible structures of graphs with the maximum number of self-loops reachable via double edge swaps we have a complete categorization of the degree sequences where double edge swaps can change any graph into any other valid graph. This understanding is exemplified in Algorithm 1, which can detect whether a degree sequence has a connected space or not. Further, we proved that augmenting double-edge swaps with triangle-loop swaps connects the space of loopy-graphs, creating the first provably correct MCMC technique for sampling loopy-graphs. In addition to filling a gap in the understanding of graph space connectivity, this work builds a tool to allow for the sampling of loopy-graphs and their subsequent use as statistical null-models. As greater emphasis is placed on sampling graphs without labeled-stubs the need for carefully sampling loopy-graphs will likely increase.

8 Acknowledgments

This work was aided by comments and suggestions from: Bailey Fosdick, Dan Larremore and Johan Ugander.

References

- [1] C Berge. Theory of graphs and its applications. *Methuen, London*, 1962.
- [2] Daniel Bienstock and Oktay Günlük. A degree sequence problem related to network design. *Networks*, 24(4):195–205, 1994.
- [3] Béla Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.
- [4] C. J. Carstens. Proof of uniform sampling of binary matrices with fixed row sums and column sums for the fast curveball algorithm. *Phys. Rev. E*, 91:042812, Apr 2015.
- [5] CJ Carstens and KJ Horadam. Switching edges to randomize networks: what goes wrong and how to fix it. *Journal of Complex Networks*, page cnw027, 2016.
- [6] Corrie Jacobien Carstens, Annabell Berger, and Giovanni Strona. Curveball: a new generation of sampling algorithms for graphs with fixed degree sequence. *arXiv preprint arXiv:1609.05137*, 2016.
- [7] Camil Demetrescu, Andrew Goldberg, and David Johnson. 9th dimacs implementation challenge—shortest paths. *American Mathematical Society*, 2006.
- [8] RB Eggleton and Derek Allan Holton. *The graph of type $(0, \infty, \infty)$ realizations of a graphic sequence*. Springer, 1979.
- [9] RB Eggleton and Derek Allan Holton. *Simple and multigraphic realizations of degree sequences*. Springer, 1981.
- [10] B. Fosdick, D. Larremore, J. Nishumura, and J. Ugander. Configuring random graph models with fixed degree sequences. *preprint*, <http://arxiv.org/abs/1608.00607>, 2016. preprint, <http://arxiv.org/abs/1608.00607>.
- [11] SL Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph ii. uniqueness. *Journal of the Society for Industrial & Applied Mathematics*, 11(1):135–147, 1963.
- [12] Takashi Ito, Tomoko Chiba, Ritsuko Ozawa, Mikio Yoshida, Masahira Hattori, and Yoshiyuki Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences*, 98(8):4569–4574, 2001.
- [13] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2, 2007.
- [14] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

- [15] Mark EJ Newman, Steven H Strogatz, and Duncan J Watts. Random graphs with arbitrary degree distributions and their applications. *Physical review E*, 64(2):026118, 2001.
- [16] Julius Petersen. Die theorie der regulären graphs. *Acta Mathematica*, 15(1):193–220, 1891.
- [17] Shai S Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1):64–68, 2002.
- [18] Richard Taylor. *Constrained switchings in graphs*. Springer, 1981.
- [19] Haiyuan Yu, Pascal Braun, Muhammed A Yildirim, Irma Lemmens, Kavitha Venkatesan, Julie Sahalie, Tomoko Hirozane-Kishikawa, Fana Gebreab, Na Li, Nicolas Simonis, et al. High-quality binary protein interaction map of the yeast interactome network. *Science*, 322(5898):104–110, 2008.
- [20] C Dieter Zander, Neri Josten, Kim C Detloff, Robert Poulin, John P McLaughlin, and David W Thieltges. Food web including metazoan parasites for a brackish shallow water ecosystem in germany and denmark. *Ecology*, 92(10):2007–2007, 2011.
- [21] G Zhang. Traversability of graph space with given degree sequence under edge rewiring. *Electronics letters*, 46(5):351–352, 2010.